DETAILED ACTION

Restriction election

1. Applicant's response (received on 04/19/2010) to Office's Requirement for Restriction/Election (mailed on 03/17/2010) is reviewed. The Requirement for Restriction/Election is withdrawn. Claims 1-80 are currently pending and under consideration.

2. Amendment received on 05/05/2008 has been entered into record. Claims 38-55 are amended. Claims 1-17 and 56-80 are withdrawn. Claims 1-80 are currently pending.

Priority

3. This application has no priority claim made. The filing date is 08/18/2003.

Examiner's Amendment

4. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

5. Authorization for this examiner's amendment was given in a telephone interview with Robert C. Kowert (reg. no. 39,255) on 06/18/2010.

6. The application has been amended against applicant submitted claim set dated 05/05/2008 as follows (only examiner amended claims are shown):

IN THE CLAIMS

1. (Currently amended) A Web Service system, comprising:

a plurality of heterogeneous hardware components configured to implement a Web Service providing one or more services, wherein the Web Service comprises:

a service provider configured to provide access to the one or more services ~~on~~ provided by the Web Service ~~system~~ via a network; and

one or more service requesters configured to access the one or more services ~~from~~ via the service provider ~~via a~~ over the network;

wherein the Web Service system is configured and implemented according to a vendor-independent Web Service architecture ~~framework for designing Web Services comprising a plurality of heterogeneous components in accordance with~~ generated according to a structured ~~methodology and one or more design patterns~~ design process for designing and generating vendor-independent Web Service architectures such that:

the plurality of heterogeneous hardware components are organized according to two or more tiers and two or more layers of the Web Service architecture; and

one or more Web Services design patterns are applied to the Web Service architecture, wherein each design pattern models a particular structure that is applicable to the Web Service.

2. (Currently amended) The Web Service system as recited in claim 1, wherein the Web Service architecture ~~framework~~ is configured to incorporate Quality of Services including reliability, scalability, and availability on the Web Service system.

3. (Currently amended) The Web Service system as recited in claim 1, wherein the Web Service further ~~comprising~~ comprises a service broker configured to interact with the service provider and service requester to negotiate and provide the services ~~of the service provider~~ to the service requester.

5. (Currently amended) The Web Service system as recited in claim 1, wherein the Web Service system further ~~comprising~~ comprises a service registry, wherein the service provider is further configured to register and publish the services in the service registry, and wherein the service requester is further configured to discover the service~~s~~ ~~provider~~ through the service registry.

6. (Currently amended) The Web Service system as recited in claim 1, wherein the Web Service system is a Business-to-Consumer Web Service system, wherein the service provider is a business service provider, and wherein the service requester is an end user.

7. (Currently amended) The Web Service system as recited in claim 1, wherein the Web Service system is a Business-to-Business Web Service system, wherein the service provider is a business service provider, and wherein the service requester is a server.

8. (Currently amended) The Web Service system as recited in claim 1, wherein the Web Service system comprises a plurality of layers as defined by the Web Service architecture, wherein the plurality of layers comprises two or more of:

a network layer configured to serve as an underlying network for services;

a transport layer for delivering messages between components of the Web Service ~~system~~;

a service description language layer configured to describe service type and functionality of the services of the service provider;

a transaction routing layer configured to route messages on the transport layer;

a service discovery layer configured to search for and locate services;

a service negotiation layer configured to negotiate exchanges between the service requesters and the service provider;

a management layer configured for provisioning of the services and for monitoring and administration of the services;

a Quality of Service layer configured to provide reliability, scalability, and availability on the Web Service system;

a security layer configured to provide authentication, entitlement, and nonrepudiation security on the transport layer; or ~~and~~

an Open Standards layer.

9. (Currently amended) The Web Service system as recited in claim 8, wherein the Open Standards layer is eXtensible Markup Language (XML).

10. (Currently amended) The Web Service system as recited in claim 8, wherein the network layer is [[the]] public Internet over Transmission Control Protocol/Internet Protocol (TCP/IP).

11. (Currently amended) The Web Service system as recited in claim 8, wherein the transport layer is one of Hypertext Transport Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and Simple Object Access Protocol (SOAP) over HTTP.

12. (Currently amended) The Web Service system as recited in claim 1, wherein the one or more Web Services design patterns include one or more of:

one or more scalability design patterns;

one or more reliability design patterns;

one or more manageability design patterns;

one or more availability design patterns; or ~~and~~

one or more security design patterns.

13. (Currently amended) The Web Service system as recited in claim 1, wherein the one or more Web Services design patterns include one or more of:

one or more Quality of Services design patterns;

one or more Integration design patterns; or ~~and~~

one or more Security design patterns.

14. (Currently amended) The Web Service system as recited in claim 13, wherein the Quality of Services design patterns include one or more of:

a Simple Object Access Protocol (SOAP) Cache Design Pattern;

a Java Message Service (JMS) Bridge Design Pattern;

a Multiple Servlet Engines Design Pattern;

an Hypertext Transport Protocol (HTTP) Load Balancer Design Pattern;

a State Management Design Pattern;

a SOAP Logger Design Pattern;

a High Availability of Service Registry Design Pattern;

a Universal Description, Discovery and Integration (UDDI) Deployment Design Pattern;

a Publish, Unpublish, and Discover Web Services Design Pattern;

a Version Management of Deployment and Service Registry Design Pattern; or ~~and~~

a Registry Content Management Design Pattern.

15. (Currently amended) The Web Service system as recited in claim 13, wherein the Integration design patterns include one or more of:

an Application-to-Application Design Pattern;

a Standard Build Design Pattern;

a Hub-Spoke Replication Design Pattern;

a Federated Replication Design Pattern;

a Multi-Step Application Integration Design Pattern;

a Data Exchange Design Pattern;

a Closed Process Integration Design Pattern;

an Open Process Integration Design Pattern;

a Service Consolidation-Broker Integration design pattern; or ~~and~~

a Reverse Auction-Broker Integration design pattern.

16. (Currently amended) The Web Service system as recited in claim 13, wherein the Security design patterns include one or more of:

A Single Sign-on Design Pattern; or ~~and~~

a Messaging Transport Design Pattern.

17. (Currently amended) The Web Service system as recited in claim 1, wherein the Web Service system comprises a plurality of tiers as defined by the Web Service architecture, wherein the plurality of tiers comprises two or more of:

a client tier;

a presentation tier;

a business tier;

an integration tier; or ~~and~~

a resource tier.

18. (Currently amended) A system for generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components, comprising:

means for generating one or more Use Cases for the Web Service in accordance with one or more design patterns, wherein each Use Case models a particular business scenario for the Web Service, and wherein each design pattern models a structure that is applicable under certain conditions;

means for generating a high-level architecture for the Web Service [[and]] in accordance with the one or more design patterns, wherein said means for generating the high-level architecture identifies two or more entities of the Web Service and [[the]] relationships and interactions among the two or more entities; and

means for generating a logical architecture for the Web Service according to the ~~use case~~ business scenarios modeled by the one or more Use Cases and in accordance with the one or

more design patterns, wherein said means for generating the logical architecture identifies two or more logical components of the Web Service and [[the]] relationships among the two or more logical components according to two or more tiers, and wherein the generated logical architecture comprises two or more layers.

20. (Currently amended) A method, comprising:

performing by one or more computers:

generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components in accordance with one or more design patterns, wherein each design pattern models a particular structure that is applicable in a Web Service under certain conditions, and wherein said generating a vendor-independent Web Services architecture comprises:

generating one or more Use Cases for the Web Service in accordance with the one or more design patterns, wherein each Use Case models a particular business scenario for the Web Service, and wherein said generating one or more Use Cases comprises:

obtaining requirements, including user requirements and technical requirements, for the Web Service; and

encapsulating the obtained requirements for the Web Service into the one or more Use Cases for the Web Service;

generating a high-level architecture for the Web Service in accordance with the one or more design patterns, wherein said generating the high-level architecture ~~identifies~~ comprises identifying two or more entities of the Web Service and [[the]] relationships and interactions among the two or more entities; and

generating a logical architecture for the Web Service according to the ~~use case~~

business scenarios modeled by the one or more Use Cases and in accordance with the one or more design patterns, wherein said generating the logical architecture comprises identifying ~~identifies~~ two or more logical components of the Web Service and [[the]] relationships among the two or more logical components according to two or more tiers, and wherein the logical architecture comprises two or more layers; and

implementing the Web Service according to the generated Web Service architecture.

21. (Currently amended) The method as recited in claim 20, wherein said generating a high-level architecture for the Web Service further comprises identifying one or more Open Standards protocols for use in said interactions among the two or more entities.

23. (Currently amended) The method as recited in claim 20, wherein the two or more entities comprise:

a service provider configured to provide one or more services on the Web Service; and

one or more service requesters configured to access the one or more services from the service provider via a network.

24. (Currently amended) The method as recited in claim 23, wherein the two or more entities further comprise a service broker configured to interact with the service provider and service requester to negotiate and provide the services of the service provider to the service requester.

25. (Currently amended) The method as recited in claim 23, wherein the two or more entities further comprise a service registry, wherein the service provider is further configured to register and publish the services in the service registry, and wherein the service requester is further configured to discover the service provider through the service registry.

28. (Currently amended) The method as recited in claim 23, wherein the two or more layers of

the logical architecture comprise two or more of:

a network layer configured to serve as an underlying network for services;

a transport layer for delivering messages between components of the Web Service;

a service description language layer configured to describe service type and functionality of the services of the service provider;

a transaction routing layer configured to route messages on the transport layer;

a service discovery layer configured to search for and locate services;

a service negotiation layer configured to negotiate exchanges between the service requesters and the service provider;

a management layer configured for provisioning of the services and for monitoring and administration of the services;

a Quality of Service layer configured to provide reliability, scalability, and availability on the Web Service;

a security layer configured to provide authentication, entitlement, and non-repudiation security on the transport layer; or ~~and~~

an Open Standards layer.

29. (Currently amended) The method as recited in claim 28, wherein the Open Standards layer is eXtensible Markup Language (XML).

30. (Currently amended) The method as recited in claim 28, wherein the network layer is [[the]] public Internet over Transmission Control Protocol/Internet Protocol (TCP/IP).

31. (Currently amended) The method as recited in claim 28, wherein the transport layer is one of Hypertext Transport Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and Simple Object Access Protocol (SOAP) over HTTP.

32. (Currently amended) The method as recited in claim 20, wherein the one or more design patterns include one or more of:

one or more scalability design patterns;

one or more reliability design patterns

one or more manageability design patterns;

one or more availability design patterns; or ~~and~~

one or more security design patterns.

33. (Currently amended) The method as recited in claim 20, wherein the one or more design patterns include one or more of:

one or more Quality of Services design patterns;

one or more Integration design patterns; or ~~and~~

one or more Security design patterns.

34. (Currently amended) The method as recited in claim 33, wherein the Quality of Services design patterns include one or more of:

a Simple Object Access Protocol (SOAP) Cache Design Pattern;

a Java Message Service (JMS) Bridge Design Pattern;

a Multiple Servlet Engines Design Pattern;

an Hypertext Transport Protocol (HTTP) Load Balancer Design Pattern;

a State Management Design Pattern;

a SOAP Logger Design Pattern;

a High Availability of Service Registry Design Pattern;

a Universal Description, Discovery and Integration (UDDI) Deployment Design Pattern;

a Publish, Unpublish, and Discover Web Services Design Pattern;

a Version Management of Deployment and Service Registry Design Pattern; or ~~and~~

a Registry Content Management Design Pattern.

35. (Currently amended) The method as recited in claim 33, wherein the Integration design patterns include one or more of:

an Application-to-Application Design Pattern;

a Standard Build Design Pattern;

a Hub-Spoke Replication Design Pattern;

a Federated Replication Design Pattern;

a Multi-Step Application Integration Design Pattern;

a Data Exchange Design Pattern;

a Closed Process Integration Design Pattern;

an Open Process Integration Design Pattern;

a Service Consolidation–Broker Integration design pattern; or ~~and~~

a Reverse Auction–Broker Integration design pattern.

36. (Currently amended) The method as recited in claim 33, wherein the Security design patterns include one or more of:

a Single Sign-on Design Pattern; or ~~and~~

a Messaging Transport Design Pattern.

37. (Currently amended) The method as recited in claim 20, wherein the logical architecture further comprises a plurality of tiers, wherein the plurality of tiers comprises two or more of:

a client tier;

a presentation tier;

a business tier;

an integration tier; or ~~and~~

a resource tier.

38. (Currently amended) A non-transitory computer-accessible storage medium storing program instructions, wherein the program instructions are computer-executable to implement:

generating a vendor-independent Web Service architecture for implementing a Web Service, wherein the Web Service architecture comprises ~~comprising~~ a plurality of heterogeneous components in accordance with one or more design patterns, wherein each design pattern models a particular structure that is applicable in a Web Service under certain conditions, and wherein, in said generating a vendor-independent Web Services architecture, the program instructions are computer-executable to implement ~~comprises~~:

generating one or more Use Cases for the Web Service in accordance with the one or more design patterns, wherein each Use Case models a particular business scenario for the Web Service, and wherein, in said generating one or more Use Cases, the program instructions are computer-executable to implement:

obtaining requirements, including user requirements and technical requirements, for the Web Service; and

encapsulating the obtained requirements for the Web Service into the one or more

Use Cases for the Web Service;

generating a high-level architecture for the Web Service in accordance with the one or more design patterns, wherein, in said generating the high-level architecture, the program instructions are computer-executable to implement ~~identifies~~ identifying two or more entities of the Web Service and [[the]] relationships and interactions among the two or more entities;

generating a logical architecture for the Web Service according to the ~~use case~~ business scenarios modeled by the one or more Use Cases and in accordance with the one or more design patterns, wherein, in said generating the logical architecture, the program instructions are computer-executable to implement identifying ~~identifies~~ two or more logical components of the Web Service and [[the]] relationships among the two or more logical components according to two or more tiers, and wherein the logical architecture comprises two or more layers~~; and~~

~~implementing the Web Service according to the Web Service architecture~~.

39. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 38, wherein, in said generating a high-level architecture for the Web Service, the program instructions are further computer-executable to implement identifying one or more Open Standards protocols for use in said interactions among the two or more entities.

40. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 38, wherein the Web Service architecture incorporates Quality of Services including reliability, scalability, and availability on the Web Service.

41. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 38, wherein the two or more entities comprise:

a service provider configured to provide one or more services on the Web Service; and

one or more service requesters configured to access the one or more services from the service provider via a network.

42. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 41, wherein the two or more entities further comprise a service broker configured to interact with the service provider and service requester to negotiate and provide the services of the service provider to the service requester.

43. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 41, wherein the two or more entities further comprise a service registry, wherein the service provider is further configured to register and publish the services in the service registry, and wherein the service requester is further configured to discover the service provider through the service registry.

44. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 41, wherein the Web Service is a Business-to-Consumer Web Service, wherein the service provider is a business service provider, and wherein the service requester is an end user.

45. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 41, wherein the Web Service is a Business-to-Business Web Service, wherein the service provider is a business service provider, and wherein the service requester is a server.

46. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 41, wherein the two or more layers of the logical architecture comprise two or more of:

a network layer configured to serve as an underlying network for services;

a transport layer for delivering messages between components of the Web Service;

a service description language layer configured to describe service type and functionality of the services of the service provider;

a transaction routing layer configured to route messages on the transport layer;

a service discovery layer configured to search for and locate services;

a service negotiation layer configured to negotiate exchanges between the service requesters and the service provider;

a management layer configured for provisioning of the services and for monitoring and administration of the services;

a Quality of Service layer configured to provide reliability, scalability, and availability on the Web Service;

a security layer configured to provide authentication, entitlement, and non-repudiation security on the transport layer; or ~~and~~

an Open Standards layer.

47. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 46, wherein the Open Standards layer is eXtensible Markup Language (XML).

48. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 46, wherein the network layer is [[the]] public Internet over Transmission Control Protocol/Internet Protocol (TCP/IP).

49. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 46, wherein the transport layer is one of Hypertext Transport Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and Simple Object Access Protocol (SOAP) over HTTP.

50. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 38, wherein the one or more design patterns include one or more of:

one or more scalability design patterns;

one or more reliability design patterns

one or more manageability design patterns;

one or more availability design patterns; or ~~and~~

one or more security design patterns.

51. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 38, wherein the one or more design patterns include one or more of:

one or more Quality of Services design patterns;

one or more Integration design patterns; or ~~and~~

one or more Security design patterns.

52. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 51, wherein the Quality of Services design patterns include one or more of:

a Simple Object Access Protocol (SOAP) Cache Design Pattern;

a Java Message Service (JMS) Bridge Design Pattern;

a Multiple Servlet Engines Design Pattern;

an Hypertext Transport Protocol (HTTP) Load Balancer Design Pattern;

a State Management Design Pattern;

a SOAP Logger Design Pattern;

a High Availability of Service Registry Design Pattern;

a Universal Description, Discovery and Integration (UDDI) Deployment Design Pattern;

a Publish, Unpublish, and Discover Web Services Design Pattern;

a Version Management of Deployment and Service Registry Design Pattern; or ~~and~~

a Registry Content Management Design Pattern.

53. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 51, wherein the Integration design patterns include one or more of:

an Application-to-Application Design Pattern;

a Standard Build Design Pattern;

a Hub-Spoke Replication Design Pattern;

a Federated Replication Design Pattern;

a Multi-Step Application Integration Design Pattern;

a Data Exchange Design Pattern;

a Closed Process Integration Design Pattern;

an Open Process Integration Design Pattern;

a Service Consolidation–Broker Integration design pattern; or ~~and~~

a Reverse Auction–Broker Integration design pattern.

54. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 51, wherein the Security design patterns include one or more of:

a Single Sign-on Design Pattern; or ~~and~~

a Messaging Transport Design Pattern.

55. (Currently amended) The non-transitory computer-accessible storage medium as recited in claim 38, wherein the logical architecture further comprises a plurality of tiers, wherein the plurality of tiers comprises two or more of:

a client tier;

a presentation tier;

a business tier;

an integration tier; or ~~and~~

a resource tier.

56. (Currently amended) A method for designing and implementing a vendor-independent Web Service architecture, comprising:

performing by one or more computers:

identifying one or more logical components of the Web Service architecture according to one or more ~~Use Case~~ use case requirements for a Web Service, wherein each use case requirement specifies a particular business scenario for the Web Service;

translating the one or more use case requirements and one or more technical ~~constraints~~ requirements for the Web Service to determine a plurality of heterogeneous Web Service components for the Web Service architecture;

categorizing the Web Service components according to a Web Service architecture framework;

organizing the Web Service components according to two or more tiers and two or more layers of the Web Service architecture;

modifying one or more software components for the Web Service architecture according to one or more architecture principles for each of the one or more tiers and the one or more layers of the Web Service architecture;

applying one or more Web Services design patterns to the Web Service architecture ~~where appropriate~~, wherein each design pattern models a particular structure that is applicable to the Web Service;

implementing a Web Service according to the Web Service architecture; and

assessing [[the]] quality of services [[of]] provided by the Web Service after said implementing the Web Service.

58. (Currently amended) The method as recited in claim 56, further comprising generating the one or more ~~Use Cases~~ use case requirements for the Web Service in accordance with the one or more design patterns, wherein said generating one or more use case requirements comprises:

obtaining requirements, including user requirements and the technical requirements, for the Web Service; and

encapsulating the obtained requirements for the Web Service into the one or more use case requirements for the Web Service.

64. (Currently amended) The method as recited in claim 56, wherein the two or more layers comprise two or more of:

a network layer;

a transport layer;

a service description language layer;

a transaction routing layer;

a service discovery layer;

a service negotiation layer;

a management layer;

a Quality of Service layer;

a security layer; or ~~and~~

an Open Standards layer.

65. (Currently amended) The method as recited in claim 56, wherein the two or more tiers comprise two or more of:

a client tier;

a presentation tier;

a business tier;

an integration tier; or ~~and~~

a resource tier.

66. (Currently amended) The method as recited in claim 56, wherein the one or more Web Services design patterns include one or more of:

one or more Quality of Services design patterns;

one or more Integration design patterns; or ~~and~~

one or more Security design patterns.

67. (Currently amended) The method as recited in claim 66, wherein the Quality of Services design patterns include one or more of:

a Simple Object Access Protocol (SOAP) Cache Design Pattern;

a Java Message Service (JMS) Bridge Design Pattern;

a Multiple Servlet Engines Design Pattern;

an Hypertext Transport Protocol (HTTP) Load Balancer Design Pattern;

a State Management Design Pattern;

a SOAP Logger Design Pattern;

a High Availability of Service Registry Design Pattern;

a Universal Description, Discovery and Integration (UDDI) Deployment Design Pattern;

a Publish, Unpublish, and Discover Web Services Design Pattern;

a Version Management of Deployment and Service Registry Design Pattern; or ~~and~~

a Registry Content Management Design Pattern.

68. (Currently amended) The method as recited in claim 66, wherein the Integration design patterns include one or more of:

an Application-to-Application Design Pattern;

a Standard Build Design Pattern;

a Hub-Spoke Replication Design Pattern;

a Federated Replication Design Pattern;

a Multi-Step Application Integration Design Pattern;

a Data Exchange Design Pattern;

a Closed Process Integration Design Pattern;

an Open Process Integration Design Pattern;

a Service Consolidation–Broker Integration design pattern; or ~~and~~

a Reverse Auction–Broker Integration design pattern.

69. (Currently amended) The method as recited in claim 66, wherein the Security design patterns include one or more of:

a Single Sign-on Design Pattern; or ~~and~~

a Messaging Transport Design Pattern.

70. (Currently amended) A method for designing and implementing a vendor-independent Web Service architecture, comprising:

performing by one or more computers:

identifying and building one or more security components according to one or more ~~Use Case~~ use case requirements for a Web Service, wherein each use case requirement specifies a particular business scenario for the Web Service;

identifying one or more Web Service objects of the Web Service architecture to be protected;

defining an object relationship for security protection in the Web Service architecture;

identifying one or more associated trust domains for the Web Service architecture, security policy and strategy for the Web Service architecture, and one or more threat profiles for the Web Service architecture;

determining one or more protection schemes and measures for the Web Services objects;

applying one or more Web Services design patterns to the Web Service architecture where appropriate, wherein each design pattern models a particular structure that is applicable to the Web Service;

implementing a Web Service according to the Web Service architecture; and

assessing security levels of the Web Service according to two or more tiers of the Web Service architecture after said implementing the Web Service.

71. (Currently amended) The method as recited in claim 70, further comprising applying one or more Web Services security tools to implement the one or more protection schemes and measures for the Web Services objects.

72. (Currently amended) The method as recited in claim 70, further comprising generating the one or more ~~Use Cases~~ use case requirements for the Web Service in accordance with the one or more design patterns, wherein said generating one or more use case requirements comprises:

obtaining requirements, including user requirements and technical requirements, for the Web Service; and

encapsulating the obtained requirements for the Web Service into the one or more use case requirements for the Web Service.

77. (Currently amended) The method as recited in claim 70, wherein the Web Service architecture comprises a plurality of layers, where the plurality of layers comprises two or more of:

a network layer;

a transport layer;

a service description language layer;

a transaction routing layer;

a service discovery layer;

a service negotiation layer;

a management layer;

a Quality of Service layer;

a security layer; or ~~and~~

an Open Standards layer.

78. (Currently amended) The method as recited in claim 70, wherein the two or more tiers comprise two or more of:

a client tier;

a presentation tier;

a business tier;

an integration tier; or ~~and~~

a resource tier.

79. (Currently amended) The method as recited in claim 70, wherein the one or more Web Services design patterns include one or more of:

one or more Quality of Services design patterns;

one or more Integration design patterns; or ~~and~~

one or more Security design patterns.

80. (Currently amended) The method as recited in claim 70, wherein the one or more Web Services design patterns include one or more of:

a Single Sign-on Design Pattern; or ~~and~~

a Messaging Transport Design Pattern.

Reasons for Allowance

7. Claims 1-80 are allowed as amended above.

The following is an examiner's statement of reasons for allowance:

The closest prior arts of record issued to Jim Conallen (Building Web Applications with UML: Second Edition, October 10, 2002) fails to teach or suggest "a Web Service system, comprising: a plurality of heterogeneous hardware components configured to implement a Web Service providing one or more services, wherein the Web Service comprises: a service provider configured to provide access to the one or more services provided by the Web Service via a network; one or more service requesters configured to access the one or more services via the service provider over the network; wherein the Web Service system is configured and implemented according to a vendor-independent Web Service architecture generated according to a structured design process for designing and generating vendor-independent Web Service architectures such that: the plurality of heterogeneous hardware components are organized according to two or more tiers and two or more layers of the Web Service architecture; and one or more Web Services design patterns are applied to the Web Service architecture, wherein each design pattern models a particular structure that is applicable to the Web Service" in combination with all the elements of each independent claim as argued by Applicant (see last paragraph on page 1 of 1st paragraph on page 5 of Pre-Brief Conference request received on 10/20/2008 and 1st paragraph on page 25 through 2nd paragraph on page 39 of Amendment received on 05/05/2008). Applicant has agreed to amend the limitations of "a plurality of heterogeneous hardware components configured to implement a Web Service providing one or more services", "the Web Service system is configured and implemented according to a vendor-independent Web Service

architecture generated according to a structured design process for designing and generating vendor-independent Web Service architectures", "the plurality of heterogeneous hardware components are organized according to two or more tiers and two or more layers of the Web Service architecture" and "one or more Web Services design patterns are applied to the Web Service architecture, wherein each design pattern models a particular structure that is applicable to the Web Service". Examiner has reviewed the claim rejections and applied prior art, i.e. Conallen as per Office Action mailed on 08/19/2008 with respect the currently amended and claimed invention. Examiner has preformed further search based on the currently amended claim language. Examiner has also ordered and reviewed a Fast and Focus Search on the currently amended independent claim 1 language. Examiner has determined that together the closest art of Conallen does not fully disclose the limitation combination with reasonable obviousness and/or motivation. Independent claims 1, 18, 20, 38, 56 and 70 are to be allowed.

The dependent claims further limit the independent claims and are considered allowable on the same basis as the independent claims as well as for the further limitations set forth. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Peling A. Shaw whose telephone number is (571) 272-7968. The examiner can normally be reached on M-F 8:00 - 4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William C. Vaughn can be reached on (571) 272-3922. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the statu9s of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Peling A Shaw/
Examiner, Art Unit 2444